Vysoké učení technické v Brně

Fakulta informačních technologií

Síťové aplikace a správa sítí

2017/2018

Technická zpráva

Klient POP3 s podporou TLS

Tomáš Blažek (xblaze31) **Brno, 25. října 2017**

Obsah

1 Úvod

Elektronická pošta je v dnešní době značně využívaný druh komunikace mezi lidmi. K tomu jsou potřeba dvě oddělené části, které se spolu následně dorozumívají a to serverová část, kde se ukládají zprávy jednotlivých uživatelů a klientská část, která se připojuje k serveru a následně stahuje nebo odesílá zprávy na daný poštovní server. V tomto případě jsem se zabýval klientskou částí, která umožňuje čtení elektronické pošty skrze protokol POP3 s rozšířeními POP3s a POP3 STARTTLS. V dokumentaci je popsán návrh programu a základní informace o již zmíněných protokolech a jednotlivých příkazech, které jsou využity při implementaci. Ta je popsána v samostatné kapitole, kde je uvedena dekompozice jednotlivých částí programu a informace o nich. Na závěr je také uveden návod k použití a základní informace o programu.

2 Návrh aplikace

Aplikace by měla být schopná stáhnout zprávy uložené na serveru skrze POP3 (Post Office Protocol verze 3) a uložit je do zadaného adresáře, každou zprávu zvlášť. Po dokončení akce následně informovat uživatele na standardní výstup o počtu stažených zpráv. Uživatel při spouštění programu bude mít také možnost zadat parametry, kterými bude moct měnit funkcionalitu. Například adresář určený pro stahování zpráv, autorizační soubor nebo typ komunikace bez šifrování nebo s šifrování.

2.1 POP3

Post Office Protocol verze 3 určuje množinu pravidel a příkazů, které jsou potřeba pro komunikaci mezi serverem a klientem. Podle registru transportních protokolu IANA[1] komunikuje zpravidla na portu s číslem 110. Existují 3 třídy stavů serveru, ve kterých se používají různé příkazy a to autorizační, transakční a aktualizační.

2.1.1 Autorizační stav

Stav, kdy se klient připojí k serveru a dostane uvítací zprávu. V tomto stavu je možnost ukončit spojení, přihlásit se pomocí kombinace dvou příkazů USER a následně PASS nebo přejít do šifrovaného stavu komunikace pomocí příkazu STLS.

Příkazy:

- **QUIT** - Ukončí spojení se serverem a převede jej do aktualizačního stavu.
- **USER** - Pošle serveru zprávu obsahující jméno uživatele.
- **PASS** - Pošle serveru zprávu obsahující heslo uživatele.

2.1.2 Transakční stav

Po úspěšném přihlášení klienta na server se přepne do transakčního stavu, ve kterém je možné provádět operace se zprávami. Možné operace jsou například vypsání počtu zpráv, obsah určité zprávy nebo mazání zpráv.

Příkazy:

- **STAT** - Vrátí při úspěšné odpovědi počet zpráv uložených na daném učtu a velikost v bytech kolik zabírají místa na serveru.
- **LIST** - Vrátí při úspěšné odpovědi seznam číslovaný od jedné , kde na prvním místě je identifikační číslo zprávy a na druhém místě její velikost v bytech. Také je možnost volat LIST s identifikačním číslem zprávy, což vrátí velikost zprávy odpovídající danému identifikačnímu číslu.
- **RETR** - Vrátí při úspěšné odpovědi velikost zprávy v bytech a obsah celé zprávy zakončený tečkou. Volá se s identifikačním číslem zprávy, která je požadována stáhnout ze serveru.
- **DELE** - Při úspěšném provedení příkazu smaže ze serveru zprávu daného identifikačního čísla.
- **NOOP** - Tento příkaz nedělá nic. Server na něj pouze odpoví kladně pokud je komunikace stále otevřená.
- **RSET** - Vrátí změny provedené na serveru během transakčního stavu.

2.1.3 Aktualizační stav

Stav, kdy serveru dojde příkaz QUIT a server provede změny, které mu byly zadány v transakčním stavu.

2.2 POP3 STARTTLS

Klasické POP3 používá komunikaci v nešifrované podobě. STARTTLS je rozšíření, které umožňuje převést nešifrovanou komunikaci na šifrovanou. Toto se může stát pouze v autorizačním stavu serveru za pomoci příkazu STLS.[2]

2.3 POP3s

Přípona "s" značí secure, což znamená, že celé spojení je šifrováno. Podle registru transportních protokolu IANA komunikuje zpravidla na portu s číslem 995.

3 Popis implementace

Program je implementovaný v jazyce C++ a popisuje práci elektronického poštovního klienta pro čtení zpráv případně možnosti smazání zpráv ze schránky. Pro komunikaci jsou využívány BSD sokety za pomoci TCP spojení se serverem. Program je dekomponován do několika fází.

3.1 Zpracování argumentů

Argumenty příkazové řádky jsou ošetřeny pomocí funkce *getopt* z knihovny unistd.h, která identifikuje jednotlivé parametry na příkazové řádce. Vytvořil jsem si globální objekt *Arguments*, do kterého zaznamenávám, jaké parametry byly zadány popřípadě i jejich hodnoty.

3.2 Navázání spojení

Pro připojení k serveru využívám funkce *getaddrinfo*, *socket* a *connect*, které jsem zakomponoval do své funkce *createConnection*. Nejdříve si připravím strukturu *addrinfo*, kterou následně inicializuji pro *sock_stream* a určím *ai_family*, že nespecifikuji, zda-li použít ipv4 nebo ipv6. Další krok je zvolit výchozí port pro komunikaci (110 pro POP3 nebo 995 pro POP3s), pokud nebyl zadán explicitně jako parametr. Přichází na řadu získání informací o server adrese pomocí *getaddrinfo* a vytvoření soketu. Pokud se soket připojí na server funkcí *connect*, je připravena komunikace. Při neúspěchu se testuje další typ připojení, dokud se nedojde na konec vázaného seznamu struktury *addrinfo*. Tato konstrukce je uvedena na manuálových stránkách[3]. Pokud nastane tato situace, program se ukončí a žádná komunikace mezi serverem a klientem nenastane. Pro případ navázání šifrovaného spojení, jsem připravil funkci *turnSocket2SSL*, která převede obyčejný soket na SSL soket a to dvěma možnými způsoby, které jsou zadány při spuštění programu. Buď převede soket rovnou, zda-li se jedná o šifrování celé komunikace nebo zašle příkaz STLS serveru, který dá signál serveru pro přechod na šifrovanou verzi a až po té převede obyčejný soket na SSL. Před samotnou šifrovanou komunikací je potřeba ještě ověřit platnost certifikátu a až poté je možné začít komunikaci.[4]

3.3 Komunikace

Pro přímou komunikaci se serverem využívám funkce z externích knihoven *send* a *recv* pro nešifrovanou komunikaci a *SSL_read* a *SSL_write* pro šifrovanou. Pro optimalizaci kódu jsem zavedl čtyři vlastní funkce, ve kterých volám již zmíněné. Jednu pro odesílání zpráv na server a tři pro přijímání odpovědí ze serveru. Jedna přijímá pouze jeden znak. To kvůli validaci odpovědi, kdy v případě znaku mínus je odpověď s informací o chybě. Některé odpovědi ze serveru bývají víceřádkové, proto jsem připravil jednu funkci pro příjímání jednořádkových zpráv a druhou pro víceřádkové zprávy.

3.3.1 Přihlášení ke schránce

Před samotným prováděním operací na serveru je nutné přihlášení ke schránce daného uživatele. To vykonávám ve funkci, které zjistí přihlašovací údaje ze souboru zadaného při spuštění programu a dále zasílá příkaz USER s přihlašovacím jménem uživatele. Pokud následuje kladná odpověď, je zaslán ještě příkaz PASS s heslem pro daný učet. Po úspěšném přihlášení je možno začít vyřizovat potřebné operace nad danou schránkou. V opačném případě se program ukončí bez provedení příslušných akcí.

3.3.2 Stahování zpráv

Implementovány jsou dvě možnosti a to stahování všech a nebo pouze nových zpráv. Nová zpráva je taková, která ještě není stažená v zadané složce u klienta. Za tímto účelem pošlu žádost serveru s příkazem UIDL, který dopomůže k identifikaci nových zpráv, jelikož vrací unikátní identifikátor reprezentující právě jednu zprávu uloženou na serveru. Identifikátor využívám ke generování jména souborů, do kterých je ukládán obsah zpráv stažených ze serveru. Nepoužívám identifikátor ve výchozí podobě z důvodu, jelikož může nabývat číselné hodnoty, ale také řetězce náhodných znaků. Některé znaky, ale nemohou být použity v názvu souboru jako například zpětné lomítko. Tento příkaz je pouze volitelný, proto je potřeba zjistit zda-li je odpověď validní. Pokud není, je potřeba zajistit jiný způsob pojmenovávání souborů se zprávami. Zvolil jsem metodu, kdy po stažení zprávy ze serveru generuji číslo podle obsahu zprávy, které je použito jako název souboru. Stahování pouze nových zpráv, ale není tak optimální, jak s využitím příkazu UIDL, kdy kontrola nových zpráv ze schránky je takřka okamžitá i pro stovky zpráv uložených ve schránce. Po stažení zprávy jsou ještě odstraněny zdvojené znaky "." na začátku každého řádku a jsou nahrazeny jedním tímto znakem. Toto plyne z protokolu POP3, aby nedocházelo k předčasnému ukončení stahování zprávy. [5]

3.3.3 Mazání zpráv

Operace vždy smaže všechny zprávy ze schránky při použití příslušného parametru při spuštění. Provede se jako poslední v pořadí. Tudíž zprávy ze schránky, které nejsou ještě uložené lokálně u klienta jsou nejdřív stažené a až poté je schránka promazána.

4 O programu

Program byl vytvořen na platformě Linux Ubuntu 14.04 ve vývojovém prostředí CLion od firmy JetBrains. K překladu je využíván překladač g++ verze 6.4.0 s volbou standartu C++11.

4.1 Návod na použití

Překlad aplikace je vyvolám příkazem make, který vytvoří binární soubor jménem *popcl*. Spuštění aplikace se provádí z příkazové řádky se jménem *popcl* s různými volbami pomocí parametrů. Pořadí parametrů je libovolné.

Použití:

popcl <server> [-p <port>][-T|-S [-c <certfile>][-C <certaddr>]] [-d] [-n] -a <auth_file> -o <out_dir>

Popis parametrů:

Povinné:

- <server> - Doménové jméno nebo IP adresa požadovaného zdroje. Uvádí se jako parametr bez přepínače. V případě zadání více parametrů bez přepínače, je zvolen první výskyt.
- -a <auth_file> - Vynucuje autentizaci uživatele. Konfiguračního souboru <auth_file> obsahuje přihlašovací údaje ve tvaru:
  username = jmeno
  passoword = heslo
- -o <out_dir> - Specifikuje výstupní adresář <out_dir>, do kterého program ukládá stažené zprávy.

Volitelné:

- -p <port> - Určí explicitně komunikační port jinak je zvolen výchozí port registrovaný organizací IANA (110 nebo 955 v závislosti na parametru -T).
- -h -Vytiskne nápovědu k programu. Pokud je zadaný argument -h neprovádí se nic jiného než tisk nápovědy.
- -T - Zapíná šifrování celé komunikace.
- -S - Naváže nešifrovanou komunikaci se serverem a následně pomocí příkazu STLS přejde na šifrovanou variantu protokolu.
- -c <certfile> - Definuje soubor <certfile>s certifikáty, který se použije pro ověření platnosti certifikátu SSL/TLS (použití pouze s -T nebo -S).
- -C <certaddr> - Definuje adresář <certaddr>s certifikáty, který se použije pro ověření platnosti certifikátu SSL/TLS (použití pouze s -T nebo -S).
- -d - Po stažení zpráv ze serveru vymaže obsah celé schránky.
- -n - Specifikuje, že se budou stahovat pouze nové zprávy.

Literatura

[1] IANA - Service Name and Transport Protocol Port Number Registry [online]. [cit. 2017-10-26]. Dostupné z: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?search=pop3

[2] RFC 2595-Using TLS with IMAP, POP3 [online]. 1996 [cit. 2017-10-28]. Dostupné z: https://tools.ietf.org/html/rfc2595

[3] MAN-getaddrinfo [online]. 1996 [cit. 2017-10-28]. Dostupné z: http://man7.org/linux/man-pages/man3/getaddrinfo.3.html

[4] SSL/TLS Client [online]. [cit. 2017-10-29]. Dostupné z: https://wiki.openssl.org/index.php/SSL/TLS_Client

[5] RFC 1939-Post Office Protocol -Version 3 [online]. 1996 [cit. 2017-10-29]. Dostupné z: https://tools.ietf.org/html/rfc1939